Die tägliche Google - Suche: Ohne Mathematik undenkbar!

Dipl.-Wirt.Math. Jan Maruhn

FB IV - Mathematik

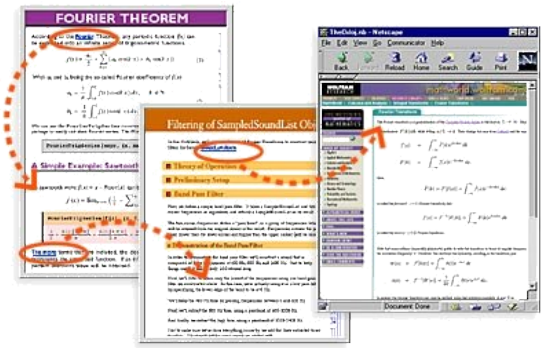
Universität Trier

29. März 2006

Gliederung

- **Einleitung und Motivation**
- Das Internet als Graph
- Rangbestimmung einer Website
- Numerisches Lösungsverfahren
- Weitere Anwendungen von LGS
- Zusammenfassung

Zur Notwendigkeit von Suchmaschinen



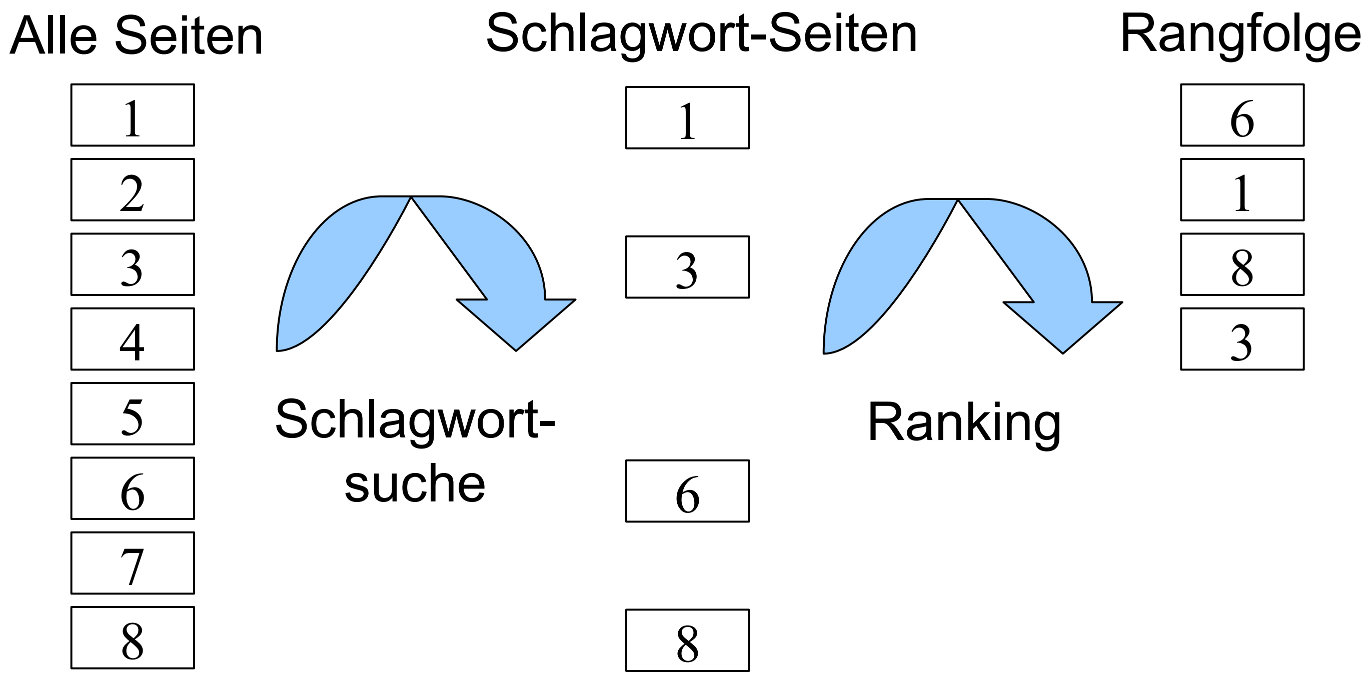
Das Internet ist eine ständig wachsende Ansammlung unstrukturierter Dokumente, die jeweils per Hyperlinks auf andere Seiten verweisen.

⇒ Um sich in diesem gewaltigen Datenbestand zurechtzufinden, sind Suchmaschinen zwingend erforderlich (z.B. AltaVista, Google, Lycos, Yahoo!,…)

⇒ Suchmaschinen folgen sukzessive der Linkstruktur und erzeugen aus den besuchten Seiten eine riesige Datenbank aufbereiteter Informationen

User-Suche in der Datenbank

Diese gewaltige Datenbank (z.B. Google: 20-30 Milliarden Seiten) kann von jedem User durchsucht werden.



Beispiel: „Miserable Failure“

Ergebnis der Google-Suchanfrage nach den Schlagwörtern „miserable failure“, zu Deutsch „miserabler Fehler“.

Fragen:

- Wie bestimmt man den Rang einer Website?
- Wie schafft Google es, den Rang so schnell zu berechnen?

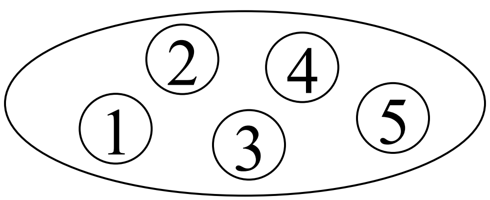
Gliederung

- Einleitung und Motivation
- **Das Internet als Graph**
- Rangbestimmung einer Website
- Numerisches Lösungsverfahren
- Weitere Anwendungen von LGS
- Zusammenfassung

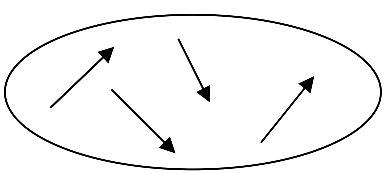
Definition eines Digraphen

Ein *Digraph* G = (V, E) besteht aus einer nichtleeren Knotenmenge V und aus einer Menge von Pfeilen $E \subset V \times V$ zwischen diesen Knoten.

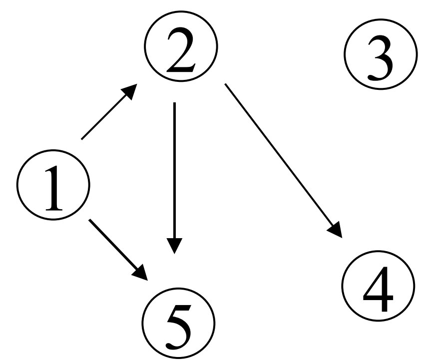
V = { 1, 2, 3, 4, 5 }



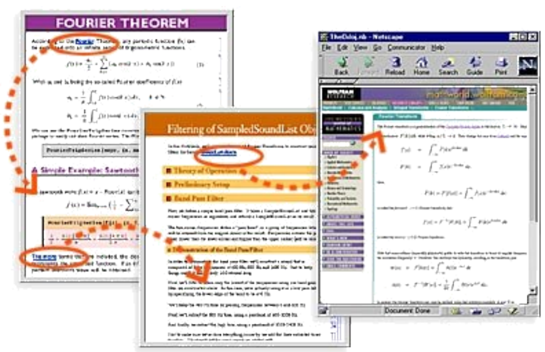
E = { (1,2), (1,5) ,(2,5), (2,4) }



Digraph G



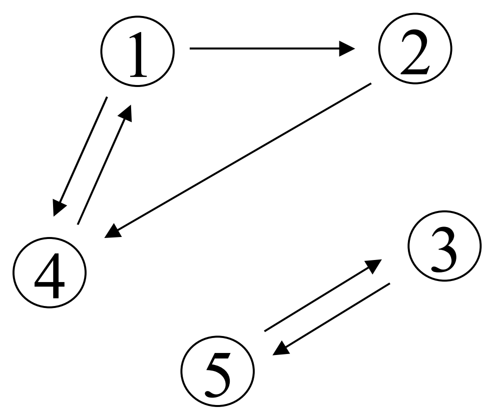
Digraph des Internets



Die Verlinkungsstruktur des Internets kann als ein riesiger Digraph betrachtet werden.

⇒ Dazu nummerieren wir die Webseiten (Knoten des Internets) beliebig und tragen die Links als Pfeile ein

1 www.tagesschau.de
2 www.phoenix.de
3 www.uni-trier.de
4 www.ard.de
5 www.ub.uni-trier.de

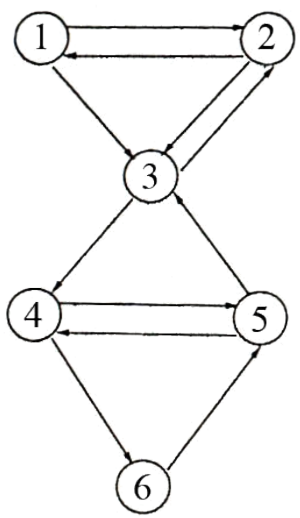


Adjazenzmatrix eines Digraphen

Die einem Digraphen G = (V,E) zugeordnete Matrix A(G) mit den Elementen

$$a_{ij} = \begin{cases} 1 & \text{, falls Website i zu Website j verlinkt} \\ 0 & \text{, falls kein Link von Website i zu Website j existiert} \end{cases}$$

heißt *Adjazenzmatrix* von G.



$$\mathbf{A(G)} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

Adjazenzmatrix eines Webs

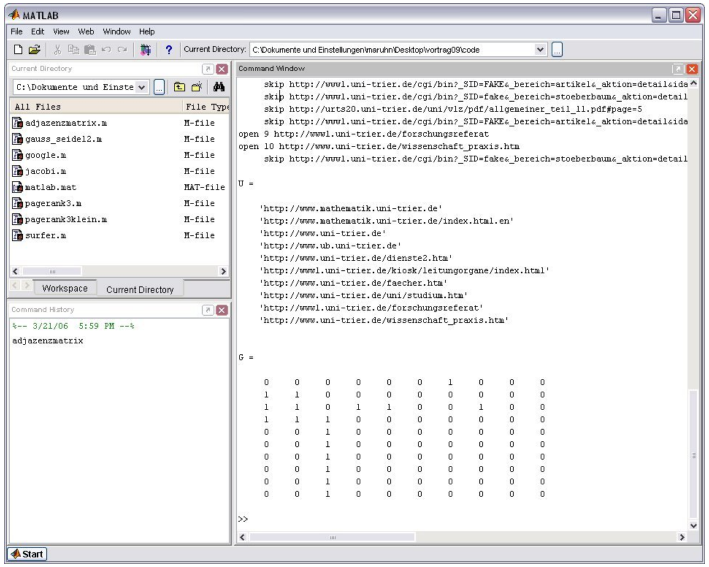
Die Verlinkungsstruktur von Websites kann vollständig durch die Adjazenzmatrix abgebildet werden.

⇒ Automatische Erzeugung dieser Matrix durch eine Software

Beispiel: Surfer-Software

Einfache Matlab-Implementierung eines Suchprogramms:

Das Programm sucht von einer Seite ausgehende Links und geht diesen bis zu einer maximalen Suchtiefe nach.



Universität Trier

Adjazenzmatrix des Internets

Die Adjazenzmatrix des Internets ist eine Matrix mit 30 Milliarden Zeilen und 30 Milliarden Spalten.

⇒ Insgesamt müssten $(30*10^9)^2 = 9*10^{20}$ Zahlen abgespeichert werden

Jedoch sind die meisten Einträge der Adjazenzmatrix gleich Null.

⇒ Speichere nur die von Null verschiedenen Elemente (ergibt noch immer mehrere hundert Milliarden Elemente)

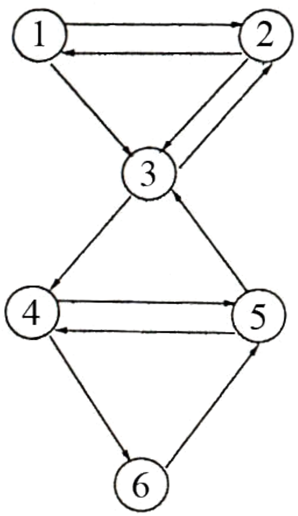
⇒ Weitere Berechnungen erfordern die Parallelisierung mehrerer hundert Computer

Gliederung

- Einleitung und Motivation
- Das Internet als Graph
- **Rangbestimmung einer Website**
- Numerisches Lösungsverfahren
- Weitere Anwendungen von LGS
- Zusammenfassung

Erste Idee eines Rankings

Je mehr Seiten auf eine Seite verlinken, umso wichtiger ist sie.



$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

$$\sum \quad \mathbf{1} \quad \mathbf{2} \quad \mathbf{3} \quad \mathbf{2} \quad \mathbf{2} \quad \mathbf{1}$$

$\Rightarrow$ Seite 3 ist am Wichtigsten, da die meisten Seiten auf sie zeigen

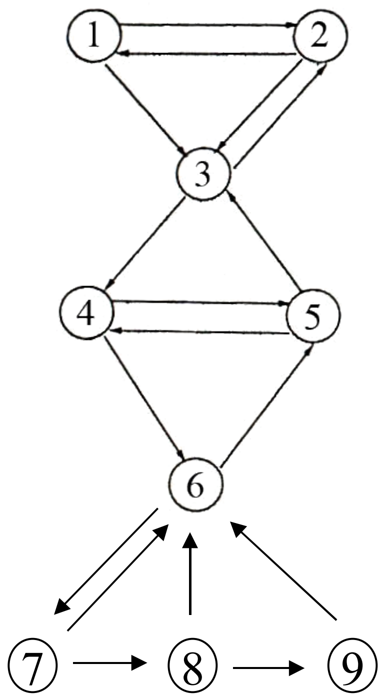
Problem: Diese Art der Rangbestimmung ist leicht manipulierbar.

Einfaches Manipulations-Beispiel

Der Inhaber von Seite 6 erstellt drei weitere Websites, die alle auf seine Website 6 verweisen.

⇒ Gemessen an der Anzahl der Links, ist nun Seite 6 am Wichtigsten

⇒ Dieses Ranking-Konzept ist in der Praxis unbrauchbar



Alternative Idee:

Nicht nur die Anzahl der Links, sondern auch deren Qualität ist entscheidend für den Rang einer Seite

Der „Page Rank“

Erfinder: Larry Page und Sergey Brin
Entwicklung des Page Rank-Verfahrens während ihres Studiums an der Stanford University (1999).

Konzept:

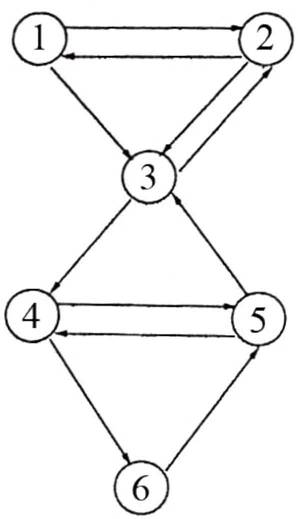
- Methode zur Kalkulation der Relevanz eines Dokuments auf Basis der Anzahl und Qualität von Linkverweisen
- Jeder Website wird eine eindeutige positive Zahl zugeordnet, welche die Wichtigkeit der Seite ausdrückt

Website	Wichtigkeit
1 www.tagesschau.de	$X_1 = ?$
2 www.phoenix.de	$X_2 = ?$
……	……

Page Rank: Rekursive Definition

Sei x_i die Wichtigkeit der Website i, i=1,…,n.

Idee: Die Wichtigkeit x_i der Website i ist die Summe der Wichtigkeiten aller auf sie verweisenden Websites, gewichtet mit der Anzahl der jeweils ausgehenden Links.



$$\left.\begin{aligned}
x_1 &= x_2/2 \\
x_2 &= x_1/2 + x_3/2 \\
x_3 &= x_1/2 + x_2/2 + x_5/2 \\
x_4 &= x_3/2 + x_5/2 \\
x_5 &= x_4/2 + x_6/1 \\
x_6 &= x_4/2
\end{aligned}\right\}$$

Homogenes lineares Gleichungs-system mit 6 Variablen

LGS mittels Adjazenzmatrix (1)

$$\mathbf{A} = \begin{array}{c} \\ \left(\begin{array}{cccccc} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right) \end{array} \begin{array}{c} \boldsymbol{\Sigma} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{1} \end{array}$$

Division der Zeilen durch Σ

$$\mathbf{B} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{P} = B^T = \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \end{pmatrix}$$

Zeilen werden
Zu Spalten

LGS mittels Adjazenzmatrix (2)

Mit der Matrix P kann das LGS geschrieben werden als:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \mathbf{P} \cdot \mathbf{x}$$

Zum Vergleich:

$$\begin{aligned}
x_1 &= x_2/2 \\
x_2 &= x_1/2 + x_3/2 \\
x_3 &= x_1/2 + x_2/2 + x_5/2 \\
x_4 &= x_3/2 + x_5/2 \\
x_5 &= x_4/2 + x_6/1 \\
x_6 &= x_4/2
\end{aligned}$$

Formale Definition des Page Rank

Seien n Websites gegeben, deren Verlinkungsstruktur durch die Adjazenzmatrix A repräsentiert wird. Die Matrix P erzeuge man durch dividieren der Zeileneinträge durch die jeweilige Zeilensumme und anschließendes Vertauschen der Zeilen und Spalten:

$$\mathbf{P} = (p_{ij}) := \big(a_{ji}/(a_{j1} + a_{j2} + ... + a_{jn})\big)_{1 \leq i,j \leq n}$$

Dann ist der *Page Rank* Lösung des Gleichungssystems Px=x.

$\Rightarrow$ In der Mathematik ist dieses spezielle lineare Gleichungssystem als „Eigenvektorgleichung“ bekannt

Modifikation der Matrix P

Um zu simulieren, dass ein Surfer auch zufällig von einer Seite zu einer beliebigen anderen springen kann, modifiziert Google die Matrix P, indem alle Einträge der Matrix mit 0,85 multipliziert werden und anschließend 0,15/n hinzuaddiert wird:

$$G = 0.85 * \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \end{pmatrix} + \frac{0.15}{6} * \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$\Rightarrow$ Ergebnis: Google-Matrix G und das zugehörige lineare Gleichungssystem Gx = x.

Eigenschaften des Gleichungssystems

- Das Gleichungssystem hat unendlich viele Lösungen, aber nur eine einzige Lösung mit $x_1+x_2+\ldots+x_n = 1$

  $\Rightarrow$ Füge diese Gleichung einfach zu dem Gleichungssystem hinzu und löse das lineare Gleichungssystem

  $$x = G^*x$$

  $$1 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

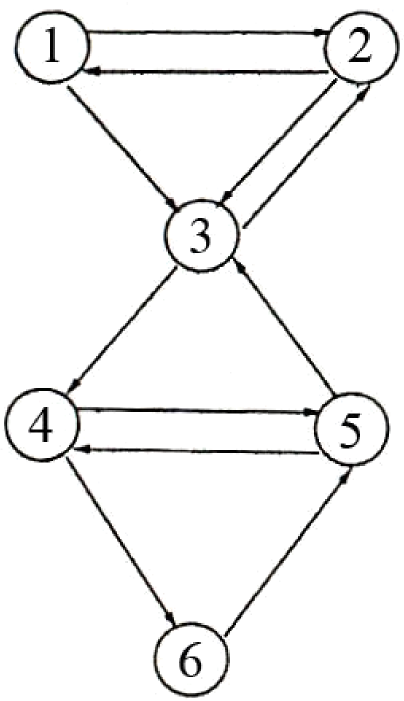
  das äquivalent umgeformt werden kann zu

$$\frac{0.15}{n}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} = \left\{\begin{pmatrix}1 & 0 & & \\ 0 & 1 & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & 1\end{pmatrix} - 0.85 \cdot P\right\} x$$

- Die resultierenden Wichtigkeiten x_i sind größer als Null

Beispiel: Kleines Web

Lösung des Gleichungssystems G*x = x mit der Zusatzbedingung $x_1+x_2+\ldots+x_n = 1$.



Seite	Page-Rank	Rang
1	0.0920	6
2	0.1577	4
3	0.2201	1
4	0.2076	3
5	0.2094	2
6	0.1132	5

Rangfolge, in der Google Suchergebnisse ausgibt

⇒ Seite 3 ist am wichtigsten

Anwendung auf den Google-Fall

Zur Bestimmung der Ränge aller Websites muss Google ein lineares Gleichungssystem mit 20-30 Milliarden Variablen lösen.

⇒ Auf Grund dieser großen Problemdimension wird der Page Rank nur einmal monatlich berechnet

⇒ Dennoch sind zur Lösung hunderte von Computern notwendig (Parallelrechner)

Insbesondere werden effiziente numerische Verfahren zur Lösung linearer Gleichungssysteme dringend benötigt.

Gliederung

- Einleitung und Motivation
- Das Internet als Graph
- Rangbestimmung einer Website
- **Numerisches Lösungsverfahren**
- Weitere Anwendungen von LGS
- Zusammenfassung

Problemstellung

Gegeben sei ein lineares Gleichungssystem der Form

$$Ax = b \iff \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$\iff \begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n & = & b_n \end{array}$$

$\Rightarrow$ Wie kann dieses möglichst schnell nach x aufgelöst werden?

Aus der Schule bekannt

Einfachstes Verfahren:

Auflösen einer Gleichung nach einer Variablen und Einsetzen in die übrig gebliebenen Gleichungen, z.B.

$$\begin{aligned} x_1 + \ \ x_2 \qquad\quad &= 1 \\ -x_1 + x_2 + x_3 &= 2 \\ x_2 + x_3 &= 4 \end{aligned} \quad \xrightarrow{\text{Auflösen nach } x_3} \quad x_3 = 4 - x_2$$

Einsetzen in Gleichung 1,2

$$\Longrightarrow \quad \begin{aligned} x_1 + \ \ x_2 \qquad\qquad\quad &= 1 \\ -x_1 + x_2 + 4 - x_2 &= 2 \end{aligned} \quad \xrightarrow{\text{Vereinfachen}} \quad -x_1 + 4 = 2$$

$$\Longrightarrow \quad x_1 = 2 \quad \xrightarrow{\text{Einsetzen in Gleichung 1}} \quad x_2 = -1 \quad \xrightarrow{\text{Einsetzen}} \quad x_3 = 5$$

Problem bei großen Gleichungssystemen

Das „Einsetz-Verfahren“ dauert bei 30.000.000.000 Gleichungen selbst für die Gewinner des Landeswettbewerbs Mathematik einfach zu lange!

⇒ Auch hunderte von Computern würden zu lange brauchen, um mittels des Einsetz-Verfahrens ein so großes Gleichungs-system zu lösen

Gibt es alternative Verfahren zur Lösung von linearen Gleichungs-Systemen, welche die Lösung schneller finden?

⇒ Ja!!! → Forschungsgebiet „Numerische Mathematik“

Alternative Idee

Anstatt das Gleichungssystem direkt durch Einsetzen zu lösen, erzeuge fortlaufend vorläufige Lösungen, welche sich der tatsächlichen Lösung immer weiter annähern.

Wie erzeugt man solche Näherungslösungen?

Wenn x das Gleichungssystem lösen würde, dann gilt für x:

$$\begin{array}{rcl} x_1 + x_2 & = & 1 \\ -x_1 + x_2 + x_3 & = & 2 \\ x_2 + x_3 & = & 4 \end{array} \quad \overset{\text{Auflösen nach } x_i}{\Longleftrightarrow} \quad \begin{array}{rcl} x_1 & = & 1 - x_2 \\ x_2 & = & 2 + x_1 - x_3 \\ x_3 & = & 4 - x_2 \end{array}$$

Idee: Starte mit einem beliebigen x und setze dieses rechts ein!

Universität Trier

Beispiel

Starte mit $x^0 = (0, 0, 0)$.

$\Rightarrow$ Einsetzen von $x^0 = (0, 0, 0)$ auf der rechten Seite

$$\left.\begin{array}{lllllll} x_1 & = & 1 - x_2 & = & 1 - 0 & = & 1 \\ x_2 & = & 2 + x_1 - x_3 & = & 2 + 0 - 0 & = & 2 \\ x_3 & = & 4 - x_2 & = & 4 - 0 & = & 4 \end{array}\right\} \text{Neues } x^1$$

$\Rightarrow$ Dieses $x^1 = (1, 2, 4)$ wieder auf der rechten Seite einsetzen

$$\left.\begin{array}{lllllll} x_1 & = & 1 - x_2 & = & 1 - 2 & = & -1 \\ x_2 & = & 2 + x_1 - x_3 & = & 2 + 1 - 4 & = & -1 \\ x_3 & = & 4 - x_2 & = & 4 - 2 & = & 2 \end{array}\right\} \text{Neues } x^2$$

Beispiel (Fortsetzung)

⇒ Einsetzen von $x^2 = (-1, -1, 2)$ auf der rechten Seite

$$\left.\begin{array}{rclcrcr} x_1 & = & 1 - x_2 & = & 1-(-1) & = & 2 \\ x_2 & = & 2 + x_1 - x_3 & = & 2-1-2 & = & -1 \\ x_3 & = & 4 - x_2 & = & 4-(-1) & = & 5 \end{array}\right\} \text{Neues } x^3$$

Nach 3 Schritten hat das Verfahren die Lösung des linearen Gleichungssystems gefunden!

⇒ Bei der Google-Matrix (30.000.000.000 Zeilen) benötigt das Verfahren weniger als 100 Iterationen, um die Lösung mit genügend Genauigkeit zu finden

Jacobi-Verfahren

Wähle einen Startvektor $x^0 = (0, 0, 0)$.

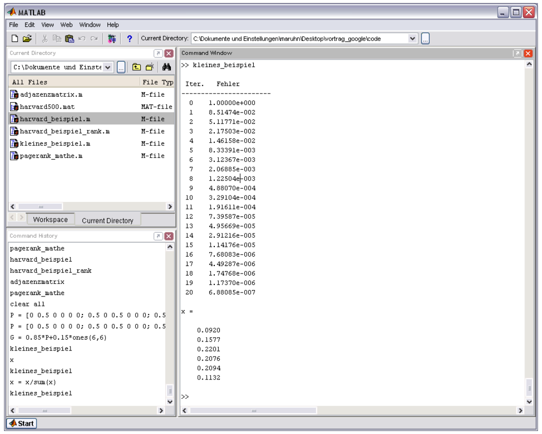
Für k=0,1,2,3,… berechne

1. Setze x^k jeweils in die rechte Seite des nach x_i aufgelösten Gleichungssystems ein
$$x_i^{k+1} = \left(b_i - a_{i,1}x_1^k - \ldots - a_{i,i-1}x_{i-1}^k - a_{i,i+1}x_{i+1}^k - \ldots - a_{i,n}x_n^k\right)/a_{i,i}$$

2. Berechne den Fehler $f = Ax^{k+1} - b$
Falls f nur kleine Einträge hat, STOP
Ansonsten setze $k \leftarrow k+1$ und gehe zu Schritt 1

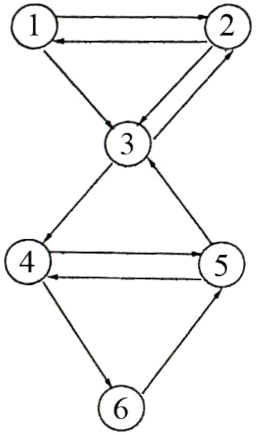
$\Rightarrow$ Dieses Verfahren funktioniert nicht bei jedem Gleichungs-system, aber immer für die Google-Systeme!

Beispiele



Testbeispiel 1:

6 Websites,
Lösung mittels
Jacobi-Verfahren
in 20 Iterationen



Testbeispiel 2: Harvard-Website mit 500 Seiten

Lösung in 29 Iterationen

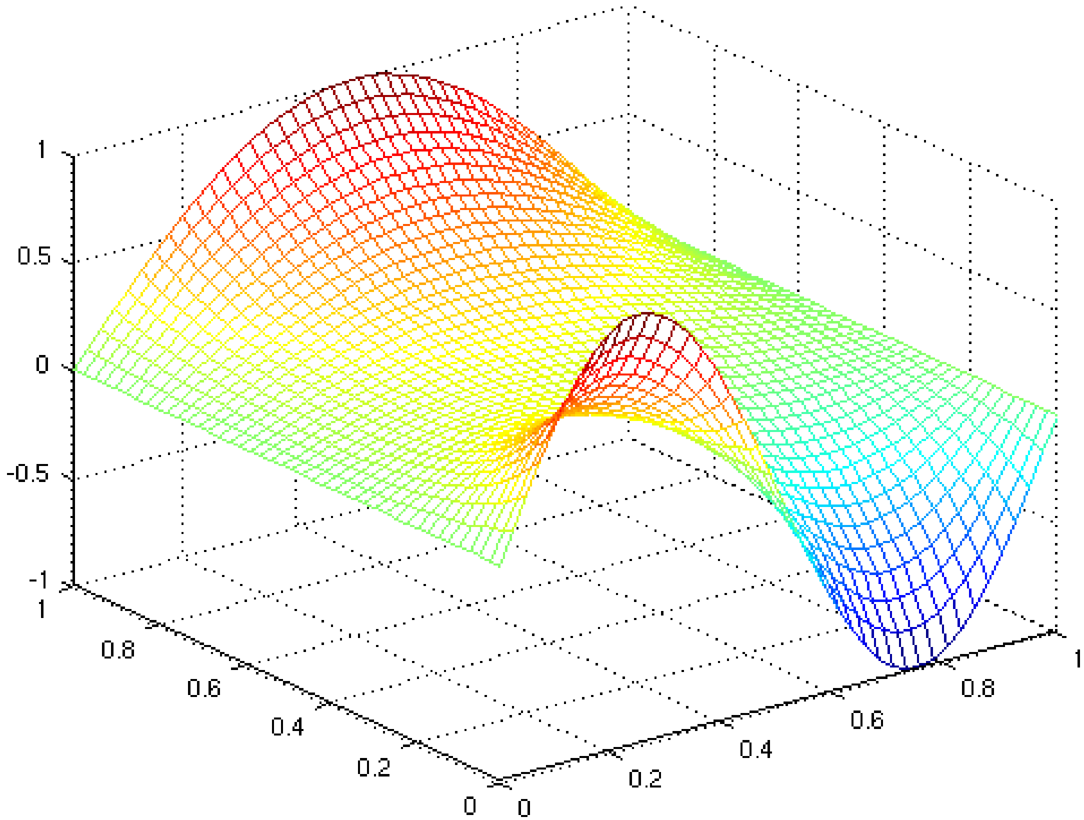
Hinweis:

Es gibt noch wesentlich bessere Verfahren zur Lösung von linearen Gleichungssystemen.

Gliederung

- Einleitung und Motivation
- Das Internet als Graph
- Rangbestimmung einer Website
- Numerisches Lösungsverfahren
- **Weitere Anwendungen von LGS**
- Zusammenfassung

Wärmeleitung von Flüssigkeiten

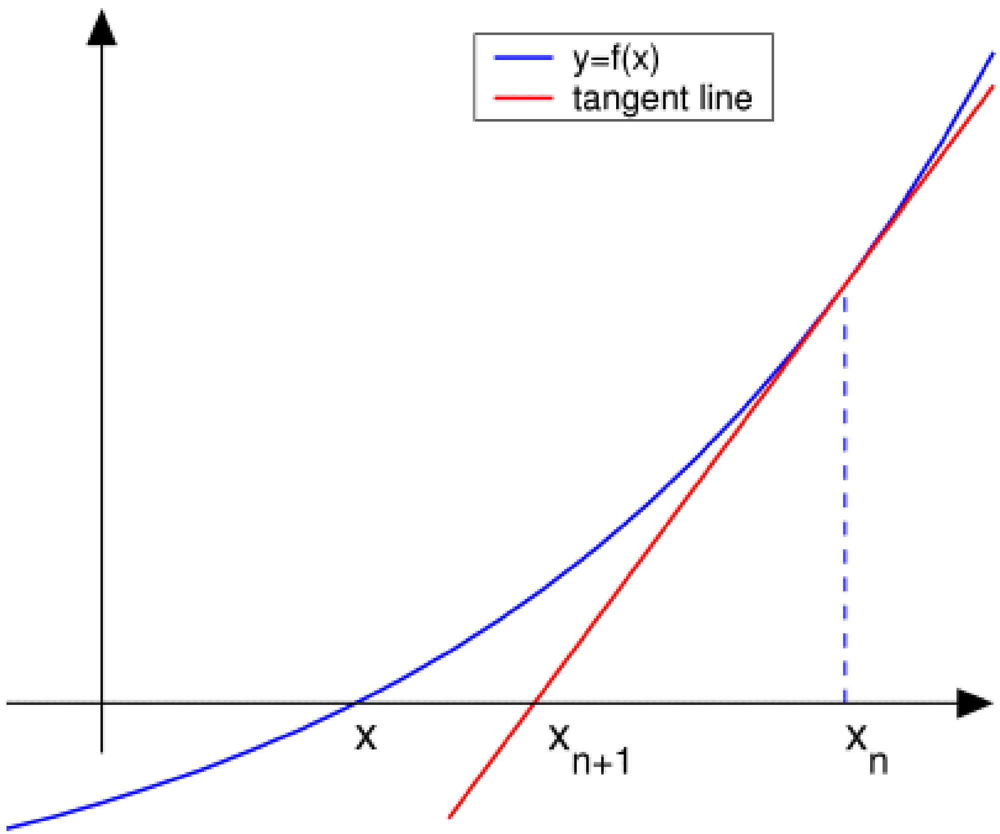


Viele Vorgänge in der Physik können durch Lösung linearer Gleichungssysteme

$$Ax = b$$

simuliert werden.

Nullstellen-Suche



Suche x mit

$$F(x) = 0$$

über

$$F'(x)s = -F(x)$$

$$x \leftarrow x + s$$

Auch bei kompli-zierten Problemen per Computer möglich!!

Gliederung

- Einleitung und Motivation
- Das Internet als Graph
- Rangbestimmung einer Website
- Numerisches Lösungsverfahren
- Weitere Anwendungen von LGS
- **Zusammenfassung**

Zusammenfassung

- Die Rangbestimmung von Webseiten führt auf riesige lineare Gleichungssysteme (20-30 Milliarden Variablen)
- Ohne schnelle numerische Verfahren ist eine Lösung des Gleichungssystems nicht möglich
- Trotz Jacobi-Verfahren ist der Rechenaufwand so groß, dass Google den PageRank nur monatlich aktualisiert
- Auch in hundert Jahren können Anwendungsprobleme noch jeden Supercomputer überfordern

Mathematiker werden dringend gebraucht!

Vielen Dank für die Aufmerksamkeit